Random Music Box

Written By: Kevin Weekly

TOOLS:

- Digital multimeter (1)
- PIC programmer (1)
- Soldering materials (1)

PARTS:

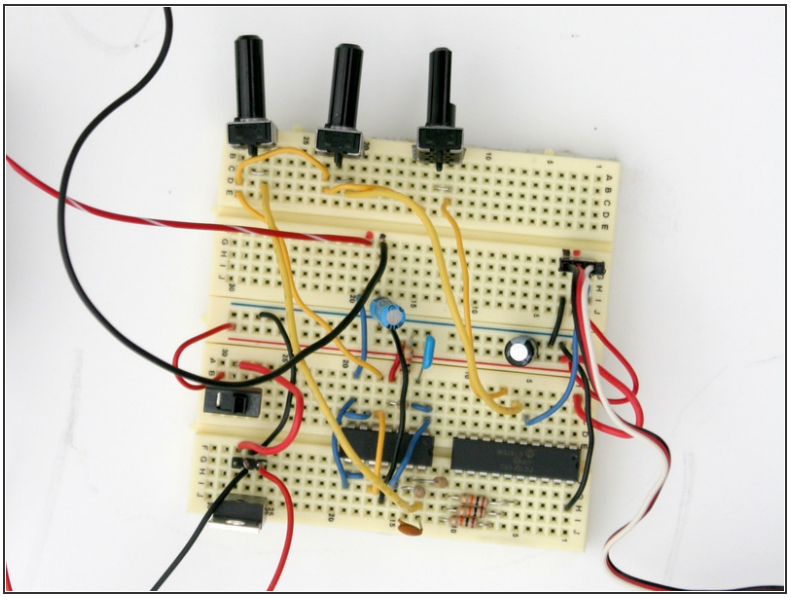
- PIC16F685 microprocessor (1)
  $3
- Op-amp chip (1)
  or use another quad op-amp, like the cheaper LM324
- Capacitor (1)
- Potentiometer (3)
  All 3 must be identical
- Resistor (6)
- Speaker (1)
- Male header (1)
- DIP socket (1)
- Hook-up wire (1)
- Breadboard (2)
  or ProtoBoard
- Power supply (1)
  I used a 4xAA battery pack with a voltage regulator.
- Masking tape (1)
- Drum and drumstick (1)
  or equivalent. For one version, I used a metal rod and a cardboard box.

- Servo (1)
  Other servos would probably work, but they must handle PWM (pulse width modulation) input. # I tried an HS-325HB and it did nothing but twitch.

SUMMARY

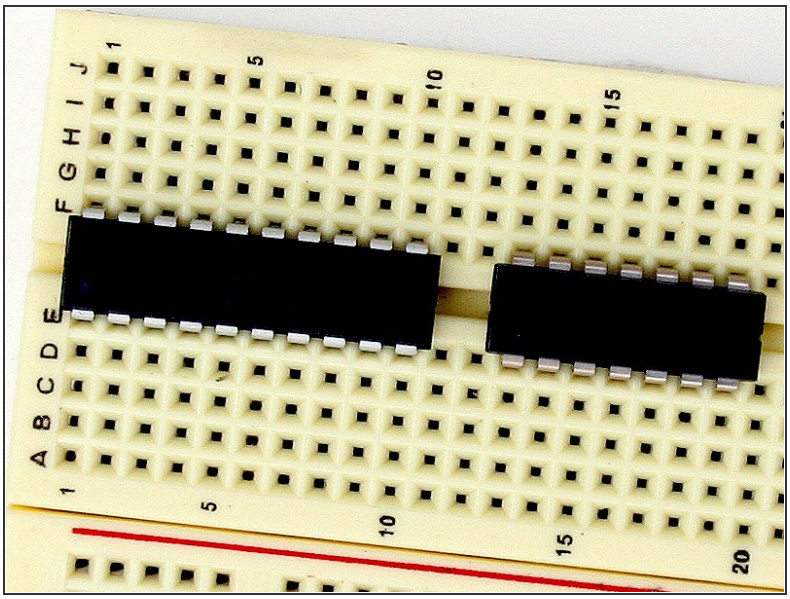
Here's a fairly inexpensive ($30–$40) project that uses a microprocessor to generate a constant stream of random but pleasant-sounding music. A Microchip PIC16F685 generates 5 square waves that are amplified and combined to play on a small speaker. A lookup table in the software stores chord progressions common in Western music. As the music runs from chord to chord, 3 oscillators play the chord itself, 1 plays a tonic-dominant (1-5) bass pattern, and 1 plays random notes from the underlying scale as a melody. Potentiometers adjust how much of each component (chord, bass, and melody) is mixed into the final output. To keep the beat, the microcontroller also generates output for driving a servomotor to strike a drum or equivalent.

Step 1 — Assemble the circuit.



- Download the project schematic at http://www.makezine.com/17/diymusic_rand....
- You can solder it onto protoboard or put it together temporarily on a solderless breadboard. I placed and connected the components in this order: sockets, resistors, capacitors, power wires, signal wires, potentiometers, and finally the off-board connections to the speaker and servo. The web page has sketches showing each step.

Step 2 — Program the microcontroller.

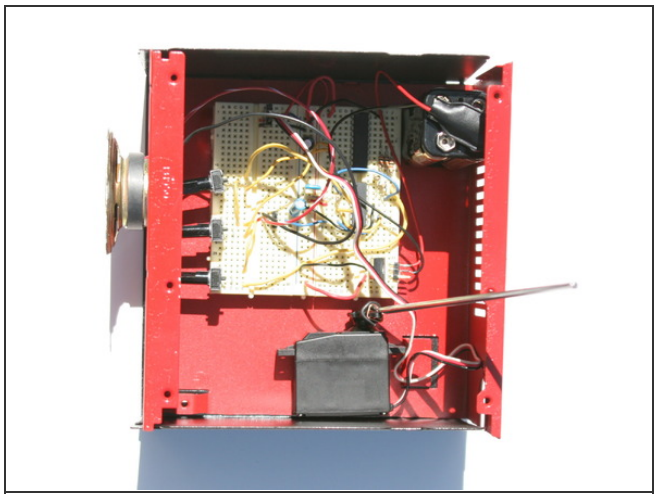


- Download the project code from makezine.com/17/diymusic_random and use your PIC programmer to burn the firmware onto the microcontroller. You can either program it directly from the hex file main.HEX or compile the program from the source code main.asm.

Step 3 — Verification

- Before inserting the ICs into the sockets and powering on, it's important to make some sanity checks, to avoid destroying the expensive ICs. First, check connections on the circuit with the digital multimeter.
- Next, make sure the power and ground rails aren't shorted out. Then power on the circuit and make sure each chip's power pin (Vcc) is getting 5V.
- Finally, plug in the ICs. If you don't hear music output, disconnect power immediately to avoid any magic smoke.

Step 4 — Add the drum.



- You should now have a noisy circuit happily playing chords. To add percussion, the servo output alternates between maximum and minimum deflection every beat. The music box MAKE built uses a metal case as both project box and drum.
- For my original version, I soldered the circuit onto protoboard and taped everything into a cardboard box.
- For those with some PIC coding experience, the source code for the firmware is mainly driven by lookup tables, which you can easily modify to do other musical things such as playing songs, scales, etc.
- For the schematic, a GIMP file with wiring, compiled firmware, source code, and an audio sample, go to http://www.makezine.com/17/diymusic_rand....

This project originally appeared in MAKE Magazine Volume 17.

Related Posts on Make: Online:

PIC-Based Melody Generator

DIY Music Roundup

This document was last generated on 2012-11-01 02:30:35 AM.